

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-289305

(43) 公開日 平成10年(1998)10月27日

(51) Int.Cl. ⁸	識別記号	F I
G 0 6 T 1/20		G 0 6 F 15/66 K
G 0 6 F 15/16	3 9 0	15/16 3 9 0 T
G 0 6 T 1/00		15/80
7/00		15/62 A
G 0 6 F 15/80		4 0 5 A

審査請求 未請求 請求項の数14 O L (全 21 頁)

(21) 出願番号 特願平9-93867

(22) 出願日 平成9年(1997)4月11日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 田村 俊之

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 東 貴裕

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(72) 発明者 田中 健一

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

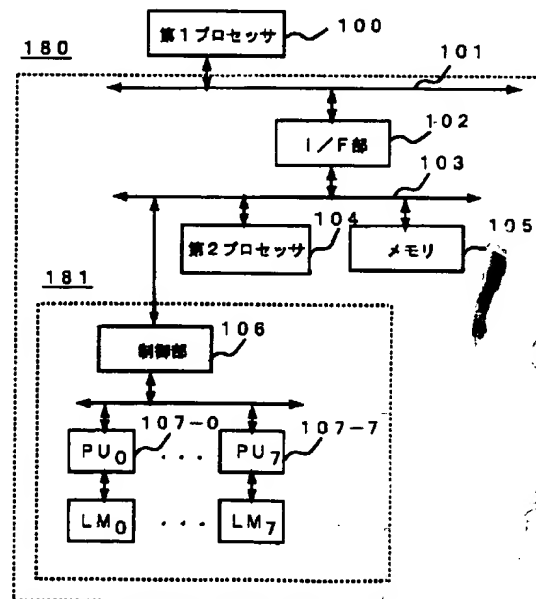
(74) 代理人 弁理士 宮田 金雄 (外2名)

(54) 【発明の名称】 画像処理装置および画像処理方法

(57) 【要約】

【課題】 高速に2つの画像を比較検証できる画像処理装置を得る。

【解決手段】 図形要素が座標で表現される画像を、該画像を予め規定した辺長の画素領域に分割し、座標表現された画像データを中間処理して中間データを作成する第1プロセッサ100、および複数の演算器107-0～107-7と該演算器の演算を制御する制御部106とを有するSIMD型プロセッサ181を含み、該第1プロセッサ100で作成された中間データを用いて、座標表現された画像データを画素値表現に変換し、第1の画像と第2の画像を比較するサブシステム180により画像処理装置を構成する。



異なる動作を行わせるようにしたことを特徴とする画像処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は2つの画像を比較し、同一画像であるかどうかを高速に検証する画像処理装置に関するものである。

【0002】

【従来の技術】従来のこの種の技術として、例えば、文献（三菱電機技報、vol.68, No.3, 1994, P.86-89）に開示されたものがある。この従来技術は、半導体のマスク作成用のデータと設計者が設計したレイアウトデータを比較し、相違箇所を検出するものであり、図20にこのような従来の画像処理装置における画像処理方法を示す。図に示すように、座標表現の半導体マスク作成用のマスクデータと設計者が設計した設計データとを入力し、入力された座標表現の画像データを画素値表現に変換する処理（S8-1）を実行する。即ち、入力された各画像データをそれぞれ、予め規定した辺長の画素領域に分割し、座標表現の各画像データを、画素内にふるパターンの占有率により、1、0のビットマップに展開することにより画素値表現に変換する。処理（S8-1）を実行した後、画素値表現の画像を比較照合する処理（S8-2）を実行し、相違部分の位置を出力する（S8-3）。従来の技術ではこれらの処理を逐次処理型の1つの計算機で行うか、あるいは複数の計算機を用意し、各画像データに対しそれぞれ1つの計算機を割り当てて、ビットマップ展開処理を画像データ毎に並列処理し、展開処理が終了後、各画像データの照合処理を行っている。

【0003】

【発明が解決しようとする課題】このように、従来の技術では各画像データのビットマップ展開処理と照合処理を逐次処理型の計算機を使用して行っているため、画像中に含まれる図形数が増大するに従って、処理時間が図形数に比例的に増大するという問題があった。また、2つの画像のビットマップ展開処理を並列に行うことは可能であったが、各々の画像のビットマップ展開処理そのものは逐次処理されており、前記と同様に、画像中に含まれる図形数が増大するに従って、処理時間が図形数に比例的に増大し、画像のもつデータ並列性を十分に活用出来ていなかった。

【0004】この発明は上記のような問題点を解消するためになされたもので、座標表現の画像データを画素値表現に変換する際においても並列処理ができ、高速に2つの画像を比較検証できる画像処理装置を得ることを目的としたものである。

【0005】

【課題を解決するための手段】この発明の第1の構成に係る画像処理装置は、図形要素が座標で表現される画像

を、該画像を予め規定した辺長の画素領域に分割し、座標表現された画像データを中間処理して中間データを作成する第1プロセッサ、および複数の演算器と該演算器の演算を制御する制御部とを有するSIMD型プロセッサを含み、該第1プロセッサで作成された中間データを用いて、座標表現された画像データを画素値表現に変換し、第1の画像と第2の画像を比較するサブシステムを備えたものである。

【0006】この発明の第2の構成に係る画像処理装置は、前記第1プロセッサによる中間処理が、頂点の座標で表現された図形を該図形の境界線を含む画素領域と、その他の画素領域に分類する処理であるものである。

【0007】この発明の第3の構成に係る画像処理装置は、前記第1プロセッサによる中間処理が、頂点の座標で表現された台形を底辺と同一方向に走査し、第1の斜辺を含む画素領域と、第2の斜辺を含む画素領域と、その他の画素領域とに分類する処理であるものである。

【0008】この発明の第4の構成に係る画像処理装置は、前記第1プロセッサによる中間処理が、頂点の座標で表現された矩形を一方方向に走査し、該矩形の境界線を含む画素領域と、その他の画素領域に分類する処理であるものである。

【0009】この発明の第5の構成に係る画像処理装置は、前記サブシステムにおいて、SIMD型プロセッサの他に第2プロセッサを備え、該第2プロセッサにより、前記画像中にある規則的に配置する複数の同一図形の繰返し表現を展開する処理を実行するようにしたものである。

【0010】この発明の第6の構成に係る画像処理装置は、前記サブシステムにおいて、SIMD型プロセッサの他に第2プロセッサと該第2プロセッサがアクセスすることの出来るメモリとを備え、該第2プロセッサは、該メモリ上に構成した領域分割テーブルを使用して画像の全部または一部を複数の区画（サブゾーン）に分割し、分割された複数の区画内にある図形要素を各区画毎にSIMD型プロセッサの各演算器に割り当てるようにしたものである。

【0011】この発明の第7の構成に係る画像処理装置は、前記SIMD型プロセッサの各演算器に前記分割された区画を割り当てる際に、第1の画像に属する区画とそれに対応する第2の画像に属する区画とを、最もデータ授受の高速な演算器の組に割り当てるようにしたものである。

【0012】この発明の第8の構成に係る画像処理装置は、前記SIMD型プロセッサとして各演算器が環状に接続されたSIMD型プロセッサを用い、第1の画像に属する区画とそれに対応する第2の画像に属する区画を隣接する演算器に割り当てるようにしたものである。

【0013】この発明の第9の構成に係る画像処理装置は、前記領域分割テーブルを複数用い、第2プロセッサ

が一方の領域分割テーブルを使用して区画分割処理を実行中に、並行して他方の領域分割テーブルを用いて前記SIMD型プロセッサに処理させるデータを供給するようにしたものである。

【0014】この発明の第10の構成に係る画像処理装置は、前記SIMD型プロセッサにおいて、各演算器が、演算結果に応じて状態が変化するフラグレジスタを個別に有しており、命令実行の際に、該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、各々の演算器毎に命令を実行するか否かを制御することができ、かつ、全フラグレジスタの値を一度に参照して、全フラグレジスタの値が更新されたか否かを判定することにより全ての演算器が予め定めた条件を満たしたか否かを判定するようにしたものである。

【0015】この発明の第11の構成に係る画像処理装置は、前記SIMD型プロセッサにおいて、各演算器が、少なくとも1個のレジスタと演算結果に応じて状態が変化するフラグレジスタとを個別に有しており、命令実行の際に、予め、画素値表現に必要な所望の座標の値を仮に決定しておき、各該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、該座標の値を更新することにより、該所望の座標を決定し、画素値表現に変換するようにしたものである。

【0016】この発明の第12の構成に係る画像処理装置は、座標で表現される画像データに含まれる台形と予め規定した辺長の画素領域との共有領域の面積を計算することにより画素値表現に変換する際に、該画素領域の境界線と該台形の一方の斜辺との交点の座標値と、該画素領域の頂点の座標値を比較した結果を条件として、予め仮に決定された座標の値を更新するようにしたものである。

【0017】この発明の第1の方法に係る画像処理方法は、複数の演算器と該演算器の演算を制御する制御部を有するSIMD型プロセッサにおいて、各演算器は、演算結果に応じて状態が変化するフラグレジスタを個別に有しており、命令実行の際に、該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、各々の演算器毎に命令を実行するか否かを制御することができ、かつ、全フラグレジスタの値を一度に参照して、全フラグレジスタの値が更新されたか否かを判定することにより全ての演算器が予め定めた条件を満たしたか否かを判定するようにしたものである。

【0018】この発明の第2の方法に係る画像処理方法は、複数の演算器と該演算器の演算を制御する制御部を有するSIMD型プロセッサにおいて、各演算器は、少なくとも1個のレジスタと演算結果に応じて状態が変化するフラグレジスタとを個別に有しており、実行命令の際に、予め、変数の値を仮に決定しておき、各該フラグ

レジスタに格納されている値が活性状態または非活性状態であることを条件として、該変数の値を更新することにより、該変数を決定し、各演算器毎に異なる動作を行わせるようにしたものである。

【0019】

【発明の実施の形態】

実施の形態1. 本実施の形態で示す画像比較システムは、前記従来例で開示している処理を従来のデータ処理プロセッサとSIMD型プロセッサを組み合わせ協調させることにより高速に実行するものである。図1に本実施の形態における画像処理装置の全体構成を示す。図1において、100は第1プロセッサ、101は汎用バス、180は第1プロセッサ100と汎用バス101で接続されたサブシステムである。サブシステム180において、102は汎用バス101とインタフェースをとるためのI/F部、103はサブシステム180内の内部バス、104、105、181は各々、内部バス103により接続された第2プロセッサ、メモリ、およびSIMD (Single Instruction stream Multiple Data stream) 型プロセッサである。SIMD型プロセッサ181は制御部(CU)106、演算器(PU)107-0~107-7で構成されている。SIMD型プロセッサ181において、CU106は、メモリを有しており、プログラムや、第1プロセッサ100、およびPU₀107-0~PU₇107-7 (以下、各PUと称する) から転送されたデータを一時的に蓄えることができる。CU106は、各PUで実行する演算実行を制御する。各PUは、それぞれローカルメモリ(LM)を有しており、CU106からの命令により演算結果を格納することが出来る。

【0020】以下に動作を説明する。まず、比較を行う対象となる画像データについて説明する。図2は該画像データの一例を示したイメージ図であり、図3は該画像データのデータフォーマットである。図2に示したように、画像データはA~Hで示される領域(以降、区画と称する)に分割されている。また、図3で示されたデータフォーマットにおいても該区画毎に図形データが記述されている。例えば、区画Aには5個の台形と1個の矩形が記述されている。なお、図形が区画の境界に跨る場合は両方の区画に図形が記述されている。また、本実施の形態では使用する画像データは、長方形(矩形)と、台形で構成されている。より詳しくは、矩形は線分(X1, Y1)(X2, Y2)を対角線とし、各辺がx軸、y軸に並行なものであり、また、台形は底辺がx軸に平行であり、4頂点が(X1, Y1)(X2, Y1)(X3, Y2)(X4, Y2)で与えられる。

【0021】このように記述された2個の画像データを比較する手順について説明する。図4は本実施の形態における画像処理システムを示す構成図である。図において、301は第1プロセッサ100上で実行する処理、

【特許請求の範囲】

【請求項1】 図形要素が座標で表現される画像を、該画像を予め規定した辺長の画素領域に分割し、座標表現された画像データを中間処理して中間データを作成する第1プロセッサ、および複数の演算器と該演算器の演算を制御する制御部とを有するSIMD型プロセッサを含み、該第1プロセッサで作成された中間データを用いて、座標表現された画像データを画素値表現に変換し、第1の画像と第2の画像を比較するサブシステムを備えたことを特徴とする画像処理装置。

【請求項2】 前記第1プロセッサによる中間処理は、頂点の座標で表現された図形を該図形の境界線を含む画素領域と、その他の画素領域に分類する処理であることを特徴とする請求項1記載の画像処理装置。

【請求項3】 前記第1プロセッサによる中間処理は、頂点の座標で表現された台形を底辺と同一方向に走査し、第1の斜辺を含む画素領域と、第2の斜辺を含む画素領域と、その他の画素領域とに分類する処理であることを特徴とする請求項2記載の画像処理装置。

【請求項4】 前記第1プロセッサによる中間処理は、頂点の座標で表現された矩形を一方方向に走査し、該矩形の境界線を含む画素領域と、その他の画素領域に分類する処理であることを特徴とする請求項2記載の画像処理装置。

【請求項5】 前記サブシステムにおいて、SIMD型プロセッサの他に第2プロセッサを備え、該第2プロセッサにより、前記画像中にある規則的に配置された複数の個の同一図形の繰返し表現を展開処理するようにしたことを特徴とする請求項1ないし4のいずれかに記載の画像処理装置。

【請求項6】 前記サブシステムにおいて、SIMD型プロセッサの他に第2プロセッサと該第2プロセッサがアクセスすることの出来るメモリとを備え、該第2プロセッサは、該メモリ上に構成した領域分割テーブルを使用して画像の全部または一部を複数の区画（サブゾーン）に分割し、分割された複数の区画内にある図形要素を各区画毎にSIMD型プロセッサの各演算器に割り当てることを特徴とする請求項1ないし5のいずれかに記載の画像処理装置。

【請求項7】 前記SIMD型プロセッサの各演算器に前記分割された区画を割り当てる際に、第1の画像に属する区画とそれに対応する第2の画像に属する区画とを、最もデータ授受の高速な演算器の組に割り当てることを特徴とする請求項6記載の画像処理装置。

【請求項8】 前記SIMD型プロセッサとして各演算器が環状に接続されたSIMD型プロセッサを用い、第1の画像に属する区画とそれに対応する第2の画像に属する区画を隣接する演算器に割り当てることを特徴とする請求項6または7記載の画像処理装置。

【請求項9】 前記領域分割テーブルを複数用い、第2

プロセッサが一方の領域分割テーブルを使用して区画分割処理を実行中に、並行して他方の領域分割テーブルを用いて前記SIMD型プロセッサに処理させるデータを供給することを特徴とする請求項6ないし8のいずれかに記載の画像処理装置。

【請求項10】 前記SIMD型プロセッサにおいて、各演算器は、演算結果に応じて状態が変化するフラグレジスタを個別に有しており、命令実行の際に、該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、各々の演算器毎に命令を実行するか否かを制御することができ、かつ、全フラグレジスタの値を一度に参照して、全フラグレジスタの値が更新されたか否かを判定することにより全ての演算器が予め定めた条件を満たしたか否かを判定するようにしたことを特徴とする請求項1記載の画像処理装置。

【請求項11】 前記SIMD型プロセッサにおいて、各演算器は、少なくとも1個のレジスタと演算結果に応じて状態が変化するフラグレジスタとを個別に有しており、命令実行の際に、予め、画素値表現に必要な所望の座標の値を仮に決定しておき、各該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、該座標の値を更新することにより、該所望の座標を決定し、画素値表現に変換することを特徴とする請求項1記載の画像処理装置。

【請求項12】 座標で表現される画像データに含まれる台形と予め規定した辺長の画素領域との共有領域の面積を計算することにより画素値表現に変換する際に、該画素領域の境界線と該台形の一方の斜辺との交点の座標値と、該画素領域の頂点の座標値を比較した結果を条件として、予め仮に決定された座標の値を更新することを特徴とする請求項11記載の画像処理装置。

【請求項13】 複数の演算器と該演算器の演算を制御する制御部を有するSIMD型プロセッサにおいて、各演算器は、演算結果に応じて状態が変化するフラグレジスタを個別に有しており、命令実行の際に、該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、各々の演算器毎に命令を実行するか否かを制御することができ、かつ、全フラグレジスタの値を一度に参照して、全フラグレジスタの値が更新されたか否かを判定することにより全ての演算器が予め定めた条件を満たしたか否かを判定するようにしたことを特徴とする画像処理方法。

【請求項14】 複数の演算器と該演算器の演算を制御する制御部を有するSIMD型プロセッサにおいて、各演算器は、少なくとも1個のレジスタと演算結果に応じて状態が変化するフラグレジスタとを個別に有しており、実行命令の際に、予め、変数の値を仮に決定しておき、各該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、該変数の値を更新することにより、該変数を決定し、各演算器毎に

302は第2プロセッサ104上で実行する処理、303はSIMD型プロセッサ181上での処理が各々示されている。より詳しくは、処理301は、ファイルを操作し、画像データをファイルから1区画分だけ読み出す処理301-1、画像データを構成する基本図形(矩形、台形)を規定する座標データを読み込み、この座標データをもとにSIMD型プロセッサ181での処理が容易になるように中間データに変換する処理(ビットマップ展開前処理)301-2、および前記中間データを汎用バス101を介して前記メモリ105に転送する処理301-3、SIMD型プロセッサ181からの画像比較データを受信する処理301-4、および比較結果を表示する処理301-5により構成されている。なお、データ転送処理301-3がメモリ105へのデータ転送を完了すると、再び、データ読みだし処理303-1を実行し、次の区画のデータを読みだし、中間データの生成処理301-2、データ転送処理301-3とファイル内に区画データが無くなるまで処理を繰り返す。この際、データの受信処理301-4、表示処理301-5の処理は前記処理301-1~301-3と逐次的に実行しても、同時並列に実行してもよい。また、サブシステム180を複数接続し、第1プロセッサから交互、もしくは処理の終了したサブシステムのメモリ105に前記中間データを転送するようにしてもよい。

【0022】302は第2プロセッサ104上で行う処理であり、前記第1プロセッサ100から転送された中間データをさらに処理し、SIMD型プロセッサ181に転送する処理、およびSIMD型プロセッサ181の動作を管理する処理とで構成されている。303はSIMD型プロセッサ181上で実行される処理を示しており、第2プロセッサ104から転送されたデータを用いてビットマップデータに展開する処理303-1、前記処理で生成された2種類のビットマップデータを各ピクセル(画素)毎に比較するビットマップ比較処理303-2、および比較処理されたデータを第1プロセッサ100に送信する処理303-3で構成されている。なお、必ずしもビットマップ比較処理は画素毎にする必要はなく、他の方法で実行してもよい。

【0023】次に、処理301、302、303が前記第1プロセッサ、第2プロセッサ、SIMD型プロセッサ上でどのように連携動作するかを説明する。第1プロセッサ、第2プロセッサ、SIMD型プロセッサ上での処理は、前記の区画を単位として行われている。本システムでは、上記の3種類のプロセッサでの処理が完了後に処理済のデータを次のプロセッサに転送し、次の区画のデータの処理を開始する。このような処理は一般にパイプライン処理と呼ばれ、並列処理の1手法として様々な分野で応用されている。

【0024】さて、本システムにおけるパイプライン動作について説明する。すなわち、第1プロセッサ100

により区画A内の図形データを中間データに変換した後、第2プロセッサ104が前記中間データの処理を開始する。このとき同時に第1プロセッサ100は次の区画、区画B内の図形データを中間データに変換する。第1プロセッサ100が区画Bの処理を、また、第2プロセッサ104が区画Aの中間データの処理を完了後、区画Aの中間データはSIMD型プロセッサ181に転送される。SIMD型プロセッサ181は、中間データのビットマップ展開処理、ビットマップデータのピクセル毎の比較処理を実行する。

【0025】次に、各プロセッサでの処理の詳細について説明する。まず、第1プロセッサ100での処理301について説明する。データ読込処理301-1はディスク上に格納された前記画像データの内の1区画分を読み込む。なお、図3に示したように本実施の形態で用いている画像データのフォーマットは各区画毎に基本図形が記述されているため、データ読込処理301-1は区画の始まりを示すフラグを読み出すことにより区画の始まりを認識でき、また、区画の終りを示すフラグを読み出すことにより当該区画内のデータの終了を検知することが出来る。

【0026】中間データへの変換処理301-2では、各頂点の座標で表現された台形、矩形の各々の基本図形のデータを基にして、SIMD型プロセッサで効率良くビットマップ展開が可能なような中間データを生成する。

【0027】以下に中間データの生成方法を説明する。なお、以降で用いる座標は画素領域の辺長を1として表現したものとする。図5は台形に対する中間データ生成の基本的な考え方を記述した図である。まず、本図に基づいて中間データ生成の概念を説明する。図中、710は台形であり、頂点711、712、713、714の座標で表現されているものである。700は平面を予め定めた分解能を単位とする正方形の領域に分割する境界線群をあらわす。上記の正方形の領域を一般にピクセル(画素)、またはピクセル領域(画素領域)と称する。701はピクセル領域の位置を指定する座標位置である。

【0028】また、図5中で使用している記号の定義を以下に示す。

(1) (X, Y)はピクセル領域の位置を示す座標値である。従って、ピクセル領域はx軸方向はXからX+1まで、y軸方向はYからY+1までの正方形の領域となる。

(2) (X1, Y1)は台形の底辺の端点でx座標の小さいものの座標。

(3) Y2は台形の上辺のy座標。

(4) IXL, IYLは左斜辺の傾き。すなわち、IXLはx軸方向の増分1に対するy軸方向の増分。IYLはy軸方向の増分1に対するx軸方向の増分であり、I

XLとIYLは互いに逆数である。ただしIYLが0のときはIXLの値に関わらず斜辺がy軸と平行であることを示すこととする。同様に、IXR、IYRは右斜辺の傾き。すなわち、IXRはx軸方向の増分1に対するy軸方向の増分。IYRはy軸方向の増分1に対するx軸方向の増分であり、IXRとIYRは互いに逆数である。ただしIYRが0のときはIXRの値に関わらず斜辺がy軸と平行であることを示す。

(5) AD1, L11, L12, L13, AD2, L21, L22, L23...は4個の整数の組(以降、単に「4つ組」と称する)(ADi, Li1, Li2, Li3)のNV個の列{(AD1, L11, L12, L13), (AD2, L21, L22, L23), ...}によって表現するのが便利である。これらの4つ組は対象とする台形の部分を含むピクセルの集まりを指定する。このとき、4つ組(AD1, L11, L12, L13)は対象とする台形の部分を含むピクセル列のうち最もy座標が小さいものを表している。ただし、ここでピクセル列とはx軸方向に並ぶ一列のピクセルである。AD1はこのピクセル列のなかで最もx座標の小さなピクセルのピクセル配列上の位置を示している。ADi(i=1, 2, 3...)は概念上は2次元配列のインデックスであるので本来は2次元値であるが、プログラム上では便宜上、1次元値であるメモリアドレスに写像した値を用いている。次に、4つ組(AD1, L11, L12, L13)の残りの3要素の意味について説明する。すなわち、AD1で指定されるアドレスのピクセルから、x軸の正方向へ並ぶL11個のピクセルを台形の左斜辺が通り、次のL12個が台形の内部にあり、その次のL13個を台形の右斜辺が通るということを示している。同様にAD2は二番目にy座標が小さいピクセル列のうち最もx座標の小さなピクセルの位置を示し、そのピクセルからx軸方向へ並ぶ台形の左斜辺が通るピクセル列の個数がL21個、台形の内部のピクセル列の個数がL22個、台形の右斜辺が通るピクセル列の個数がL23個となる。なお、本実施の形態では、4つ組をx軸の正の方向に定義したが、x軸の負の方向に定義しても同一の効果がある。

【0029】前記中間データの生成は、台形を規定する座標データから上記4つ組、IXL、IYL、IXR、IYR等を生成することに対応する。次に、この中間データのうち4つ組を生成するための手順について説明する。図6にそのフローチャートを示す。図に基づいて上記4つ組を生成するための手順について説明する。

(1) 初期化を行う(S1)。即ち、Yに本台形を含むピクセル領域のうちy座標が最も小さいY1、変数xhlにX1、変数xhrにX2を代入する。
(2) y=Yと左斜辺、および右斜辺との交点を内包するピクセル領域のx座標xll、xlrを求める(S2)。

(3) 変数xhl, xhr, xll, xlrを使用して第i番目の4つ組ADi, Li1, Li2, Li3を計算する(S3)。

(4) 変数を更新する(S4)。即ち、 $Y \leftarrow Y+1$, $xhl \leftarrow xll$, $xhr \leftarrow xlr$, $i \leftarrow i+1$ とする。

(5) 以降、(2)(3)(4)を繰り返すことにより可能なiに対する4つ組を生成することが出来る。

【0030】次に、矩形に対する中間データについて説明する。図7は中間データとして求める数値と与えられた矩形の関係を示した図である。図に示したように、矩形に対する中間データは、X, Y, N, NBvect, larg1, larg2, larg3, larg4で与えられる。以下にこれらの中間データの各要素の意味を説明する。数値larg1, larg2, larg3, larg4は矩形の領域を内包するピクセルのうち、該矩形の周を含むピクセルに該矩形領域が占める割合を示している。より詳しくは、以下の式で求めることが出来る。

$$larg1 = 1 - fr(X1)$$

$$larg2 = fr(Y2)$$

$$larg3 = fr(X2)$$

$$larg4 = 1 - fr(Y1)$$

ここで、 α を実数として、 $fr(\alpha)$ は α の小数部分を表す。変数NBvectは矩形領域を内包するピクセルのy軸方向の数。変数Nはx軸方向の数を各々示している。また、座標(X, Y)は矩形領域を内包するピクセルのうちx座標、y座標共に最も小さなピクセルの座標値である。

【0031】この様にして第1プロセッサ100において中間データ変換処理301-2が実行され、得られた中間データは転送処理301-3により、第2プロセッサに転送される。本実施の形態における第2プロセッサ104は、画像中に規則的に配置する複数個の同一図形の繰返し表現がある場合、上記中間データをさらに処理してSIMD型プロセッサ181に送るデータを作成する。

【0032】前記繰返し表現は、上で説明した基本図形を表現するための座標値に加えて、x軸方向の基本図形の繰返し数Nx、y軸方向の繰返し数Ny、x方向の繰返しの間隔dx、およびy方向の繰返し間隔dyで表現される。ここで、Nx, Nyは整数、dx, dyは実数である。すなわち、繰返し配置に属する台形は以下のように基となる基本図形を並行移動することによって表される。 $(X1+i \cdot dx, Y1+j \cdot dy)$ 、 $(X2+i \cdot dx, Y1+j \cdot dy)$ 、 $(X3+i \cdot dx, Y2+j \cdot dy)$ 、 $(X4+i \cdot dx, Y2+j \cdot dy)$ ただし、i, jは整数で、 $1 \leq i \leq Nx$, $1 \leq j \leq Ny$ を満たすものとする。なお、この場合、中間データを構成する、4つ組、IXL、IYL、IXR、IYRなどの殆どの要素は図形の形状のみに依存し、図形の位置とは

独立である。例えば、IXL、IYL、IXR、IYRは台形の斜辺の傾き、もしくは傾きの逆数を表しているため図形の位置には無関係である。また、4つ組(ADi, Li1, Li2, Li3)に関しては、第1要素ADiのみが図形の位置に依存しているだけであり、残りの要素Li1, Li2, およびLi3は図形の形状のみに依存している。したがって、繰返し図形に対する中間データは、繰返しを考えない場合の中間データの生成と、前述のNx, Ny, dx, およびdyの値に従って、4つ組の第1要素ADiに並行移動分を加算するのみで生成することが出来る。なお、以降繰返し配置を構成する図形のことを、アレイ図形と称することにする。中間データからビットマップデータに展開する際には上記のアレイ図形を展開する必要があるが、この展開処理は本実施の形態においては、第2プロセッサ104で実行される。

【0033】第2プロセッサ104は上述したアレイ図形の展開処理に加えてさらに次のような処理を実行する。この処理について説明する。本実施の形態では、図8に示すように区画を部分領域に区切って各部分領域毎に画像データを比較する。以降、この部分領域をサブゾーンと称する。なお、図3に示した画像データファイルのフォーマットでは、区画内の図形データの記述の順番は任意であり、特にこのサブゾーン毎に図形データをまとめて記述しているわけではないものとして説明する。もちろん、サブゾーン毎に図形データをまとめて記述している場合であっても本方式が適用可能であることはいうまでもない。

【0034】第2プロセッサ104は上記のような区画内の図形データをサブゾーン毎にまとめる処理をおこなう。図9は区画内の図形データをまとめる手順について説明した図である。第2プロセッサがアクセスすることの出来るメモリ105上に、領域分割テーブル401、および第1プロセッサ100からの中間データを格納する領域402を有している。テーブル401はサブゾーン毎に中間データへのポインタ(アドレス)を格納するためのエントリ、当該図形がアレイ図形を形成している場合にその位置を決定するために必要な整数値、Nx, Nyを格納する領域を有している。当該図形がアレイ図形でない場合はNx, Nyに-1を格納する。また、前記テーブル401のサブゾーンの先頭に当該サブゾーンに格納された図形数を格納するエントリがある。なお、図形の中間データを格納するメモリ領域402内には、図形の種別を示すフラグ、および当該図形がアレイ図形を形成するか否かを示すアレイフラグを含んでいる。

【0035】以下に、第2プロセッサによって図形をサブゾーン毎にまとめる方法について説明する。第2プロセッサは1区画分の中間データが格納されたメモリ領域402を先頭から読みこみ、当該する図形の座標値を取得する。この座標値から、当該の図形が占有するサブゾ

ーンを計算し、当該サブゾーンに対応する前記テーブル401のポインタを格納するエントリに、当該する図形の中間データが格納された領域の先頭アドレスを書き込むと同時に、当該サブゾーンに格納されている図形数を1増加させる。この際、もし当該図形がアレイ図形を構成している場合は、座標値を取得したのち、オフセット値Nx, Nyを用いて座標変換を予め実行し、座標変換後の座標値を以降の計算で使用する。また、このオフセット値、Nx, Nyをテーブル401に書き込む。

【0036】第2プロセッサ104はテーブル401の各エントリを読み、そのエントリに格納された中間データが実際に格納されているメモリ領域のアドレスを基に図形データを読み出すことが出来る。さらに、第2プロセッサ104は、サブゾーン毎にまとめられた図形をSIMD型プロセッサの各演算器(PU)がアクセスするローカルメモリに転送する機能を有する。

【0037】次に、読み出された中間データをローカルメモリに転送する方法について説明する。本実施の形態ではSIMD型プロセッサ181として環状に接続された8個の演算器(PU)を持つものを使用している。本実施の形態では、偶数番号をもつ4個のPUがローカルにアクセスすることが出来るローカルメモリに比較する画像のデータを、残りの奇数番号をもつPUがローカルにアクセスすることが出来るローカルメモリに基準となる画像のデータを転送する。すなわち、リングネットワークに沿って、各PUに番号PU₀, PU₁, PU₂, PU₃, . . . , PU₇を付した時に、偶数番号をもつPU₀, PU₂, PU₄, . . . のローカルメモリに1方の画像の中間データを格納し、もう一方の画像の中間データをPU₁, PU₃, PU₅, . . . に入れる。前述のように本実施の形態におけるSIMD型プロセッサは各PUを結合するネットワークがリング上になっており、隣接するPU間でのデータ転送にかかる時間が最短になっている。

【0038】本実施の形態で使用したSIMD型プロセッサ181では各PUがリング状に結合された構造を有するものを仮定したが、PU相互を結合するネットワークの形態に応じて、PUを結合する経路の短い、すなわち、PU間でのデータ転送時間が最短のPUの間でデータ比較を行うように構成すれば同一の効果が得られることは明らかである。

【0039】なお、前記領域分割テーブル401を複数使い、第2プロセッサが一方の領域分割テーブルを使用して区画分割処理を実行中に、並行して他方の領域分割テーブルを用いてSIMD型プロセッサに処理させるデータを供給するようにしてもよい。

【0040】次に、SIMD型プロセッサ181での処理について説明する。本実施の形態におけるSIMD型プロセッサ181では、第1プロセッサ100により、各々中間データに変換され、さらに、第2プロセッサに

よりサブゾーン毎にまとめられ、SIMD型プロセッサの然るべきPUのローカルメモリに格納された中間データを基に、ビットマップ展開処理、およびビットマップ比較処理を実行する。

【0041】まず、台形の中間データにおけるビットマップ展開処理について説明する。なお、本実施の形態においてはビットマップ展開処理の際は当該図形がピクセル領域に占有する割合に応じて画素値を決定する方法を採用している。図10、図11、図12は台形の中間データをビットマップに展開する手順を示したフローチャートである。本フローチャートにおいて、S5-1～S5-6は台形の左斜辺に含まれるピクセル値の計算を行う部分であり、S5-15～S5-17は台形の右斜辺に含まれるピクセル値の計算を行う部分であり、S5-7～S5-12は上記以外の部分のピクセル値を計算する部分である。

【0042】以下に、各々の部分の詳細を説明する。

(S5-1) 中間データの4つ組からADiを抽出し、サブゾーンの左下隅のピクセルのアドレスYとサブゾーンの幅Wの積Y・WをADiに加えて変数ADとし、これをADxに代入する。

(S5-2) 後述する方法により、台形の左斜辺を含むピクセルの面積を計算し、計算した結果を各PUにローカルにあるレジスタAR0の内容が指し示すローカルメモリに書き込む手順を呼び出す。

(S5-3) ADxの値に1加算し、ADxに代入するとともにレジスタAR0に代入する。

(S5-4) 減算“ $AD+Li1-ADx$ ”を実行した後、各PUにローカルに配置されたゼロフラグレジスタを参照し、もしゼロフラグがセットされていたら、すなわち、対象とするピクセルが台形の左斜辺をもちや含んでいないことを意味するが、この条件を満たすPUに限りそのローカルメモリの予め定めたアドレスに-1を書き込む(S5-5)。他方、上記の条件を満たしていないPUでは何もしない。なお、この処理は本実施の形態に用いるSIMD型プロセッサにおいて採用している条件付命令を使用している。この条件付命令の内容については後述する。

(S5-6) 全てのPUのゼロフラグ、または負フラグがセットされているか、否かを検査し、もし、全てのPUが上記条件を満足していなければ、上記の条件を全てのPUが満たすまで、S5-2～S5-6を繰り返す。なお、上記のフラグのテストは全PUで条件“ $ADx \geq AD+Li1$ ”を満たしているか否かをテストすることと等価である。

【0043】以下に前述の条件付命令について説明する。SIMD型プロセッサの特性上、各PUに異なる命令を与えることができない。本実施の形態で用いるSIMD型プロセッサは、各PUに演算結果に応じて状態が変化するローカルなフラグレジスタを有している。ま

た、このフラグレジスタの値を一度に参照する命令も具備している。前記手順(S5-4)において減算を実行した後、この演算結果に応じてフラグレジスタの内容が更新される。ゼロフラグをテストすることで、各PUにおいて、ADxが $AD+Li1$ に等しくなったか否かがわかる。さらに、本SIMD型プロセッサは各PUローカルにあるフラグレジスタを参照して、フラグがセットされていることを条件に次の命令を実行する条件付命令が具備されている。条件付命令は、もし、指定されたフラグがセットされていれば命令を実行し、セットされていなければ命令を実行しない。手順(S5-4)の場合はゼロフラグレジスタがセットされているPUに限り手順(S5-5)が実行される。なお、手順(S5-5)によって、ローカルメモリのアドレスPU番地に-1が書き込まれたPUでは以降の処理において手順(S5-2)は実行されない。すなわち、手順(S5-2)はローカルメモリのアドレスPU番地を参照し、-1が書き込まれているPUではメモリの書き込み動作は実行されない。また、以降の説明に於ても、ローカルメモリのアドレスPU番地に-1が書き込まれている場合ローカルメモリへのビットマップ展開処理は実行されないものとする。

【0044】(S5-7) 各PUのローカルレジスタAR1に1を代入する。

(S5-8) ローカルレジスタAR1の内容、すなわち1を、各ローカルレジスタAR0の指し示すローカルメモリのアドレスに書き込む。

(S5-9) ADxの値を1増加させると同時に、ローカルレジスタAR0に最新のADxの値を代入する。

(S5-10) 減算“ $AD+Li1+Li2-ADx$ ”を実行した後、各PUローカルに配置されたゼロフラグレジスタを参照し、もしゼロフラグがセットされていたら、すなわち、対象とするピクセルが台形の右斜辺を含む領域にあることを意味するが、この条件を満たすPUに限りそのローカルメモリの予め定めたアドレスに-1を書き込む(S5-11)。他方、上記の条件を満たしていないPUでは何もしない。なお、この処理は前述の条件付命令と同様の方法を使用している。

(S5-12) 全てのPUのゼロフラグ、または負フラグがセットされているか否かを検査し、もし全てのPUが上記条件を満足していなければ、上記の条件を全てのPUが満たすまで、S5-7～S5-12を繰り返す。なお、上記のフラグのテストは全てのPUが条件“ $ADx > AD+Li1+Li2$ ”を満たしているか否かを検査することと等価である。

(S5-13) 後述する台形の左斜辺を含むピクセルの面積の計算法と同様の方法により、台形の右斜辺を含むピクセルの面積を計算し、計算した結果を各PUにローカルにあるレジスタAR0の内容が指し示すローカルメモリに書き込む手順を呼び出す。

(S5-14) ADx の値に1加算し ADx に代入するとともにレジスタ ARO に代入する。

(S5-15) 減算" $AD+Li1+Li2+Li3-ADx$ "を実行した後、各PUローカルに配置されたゼロフラグレジスタを参照し、もしゼロフラグがセットされていたら、すなわち、対象とするピクセルが台形の右斜辺をもはや含んでいないことを意味するが、この条件を満たすPUに限りそのローカルメモリの予め定めたアドレスに-1を書き込む(S5-16)。他方、上記の条件を満たしていないPUでは何もしない。なお、この処理は前述の条件付命令と同様の方法を使用している。

(S5-17) 全てのPUのゼロフラグ、または負フラグがセットされているか否かを検査し、もし全てのPUが上記条件を満足していなければ、上記の条件を全てのPUが満たすまで、S5-13~S5-17を繰り返す。なお、上記のフラグのテストは全てのPUが条件" $ADx > AD+Li1+Li2+Li3$ "を満たしているか否かを検査することと等価である。

【0045】以上の手続きが完了後、S5-1に戻り、次の4つ組の処理を開始し、当該台形の間データ内に4つ組が無くなるまで処理を続ける。

【0046】以下に、前述の手順S5-2、S5-13における台形の左斜辺または右斜辺を含むピクセルの値を計算する方法を説明する。命令列は複数のPUで同時に実行される。しかし、それぞれのPUは異なるデータを扱うが、実行の分岐も全てのPUで同様に起こるため、条件分岐命令で各PU上のデータが満たす条件に応じた処理をすることはできない。ここで用いたSIMD型計算機では各PU上のデータに応じた分岐処理を「条件付命令」で実現する。「条件付命令」とは、前述したように、各PUが「条件付命令」を実行するときに、比較命令の結果、すなわち各PUに備えたフラグの内容によって、条件を満たすPUのみがその命令を実行し、その他のPUは次の命令の実行までなにもしないというものである。

【0047】以下、台形の左斜辺を含むピクセルの値を例に、SIMD型計算機で並列に複数の図形について計算する手続きの実現方法を説明する。この手続きはいくつかの2次元ベクトルで表現される台形を0から1までの実数値を要素とするピクセルの2次元配列として表現するときに用いられ、この0から1までの実数値をピクセル値と呼ぶ。ピクセルの2次元配列をピクセル配列と呼ぶことにする。ピクセルは2次元座標平面上に正方形領域を占めるものとし、この領域をそのピクセルのピクセル領域と呼ぶことにする。この手続きは2次元座標平面上に、あるピクセル領域と、ある台形が指定されたときにそのピクセルが2次元座標平面上に占める領域の面積に対して、その台形に含まれる部分の面積の割合を計算する。その割合がピクセル値の一成分となる。そして他の図形がその領域に存在する場合、予め定めた方法で

取り扱う。例えば、それらに対するピクセル値成分を加算し、ピクセル値が決定する。また、ピクセル値の最大のものをとる等の方法がある。

【0048】この手続きは計算のために変数 Xc , Y , T , YB , H , IXL , IYL , Ly , Ry , Bx , Tx を用いる。また、図13に上記手続きのフローチャートを示す。

(S11) YB と YT の初期値をピクセル領域の y 軸方向の下限、上限に設定する。即ち、 YB の値を Y とし、 YT の値を $Y+1$ とする。

(S12) 台形の上辺がピクセル領域を通る場合、 YT の値をその Y 座標にする。即ち $Y2 < YT$ を条件として YT の値を $Y2$ とする。同様に、下辺がピクセル領域を通るとき、 YB の値をその Y 座標にする。即ち、 $YB < Y1$ を条件として YB の値を $Y1$ とする。

(S13) 台形の左斜辺と、直線 $x=X1$ との交点の y 座標 Ly を定める。即ち、 Ly の値を $(IXL \cdot (X-X1) + Y1)$ とする。同様に台形の左斜辺と、直線 $x=X1+1$ との交点の y 座標 Ry を定める。即ち、 Ry の値を $(IXL \cdot (X+1-X1) + Y1)$ とする。ただし $IYL=0$ のとき、斜辺は y 軸と平行であることを示し、 $IYL=0$ を条件として Ly の値と Ry の値をもに0とする。

(S14) 図14にはピクセル領域を表わす正方形があり、これに符号709が付けてある。目的の台形がピクセル領域と交わってできる図形を表わす台形 T があり、これに符号708が付けてある。直線 $y=YT$ と、直線 $y=YB$ がピクセル領域内に作る長方形の面積から、左斜辺によって分離される目的の台形ではない部分を表わす三角形 C があり、これに符号707が付けてある。図14に示される図形 T の y 軸方向の高さ H を求めるために YT , YB を調整する。 $(Ly > Ry > YB)$ を条件として YB の値を Ry とする。そして、 $(Ly < Ry < YT)$ を条件として YT の値を Ry とする。そして H の値を $YT-YB$ によって定める。

(S15) 次に図14の図形 C の y 軸方向の高さを求めるために YT , YB を調整する。 $(YB < Ly < Ry)$ を条件として YB の値を Ly とする。そして $(Ly < Ry < YT)$ を条件として YT の値を Ly とする。

(S16) つぎに左斜辺のうち、ピクセル領域内に存在する線分の2端点の x 座標 Bx , Tx を求める。 Bx の値を $X1 - (YB - Y1) \cdot IYL$ として求め、 Tx の値を $Bx - IYL \cdot (YT - YB)$ として求める。そしてその線分の midpoint の x 座標として Xc の値を $(Tx + Bx) / 2$ とする。すると図14の図形 C の面積は $(Xc - X) \cdot (YT - YB)$ となる。

(S17) $H - (Xc - X) \cdot (YT - YB)$ が図14の図形 T の面積であり、これがこの手続きで求める結果となる。

【0049】以上手順(S11)~(S17)で求めら

れた面積を、ローカルレジスタAR0の内容が指し示すローカルメモリのアドレスに書き込んで処理は終了し、本手順が呼び出された次の命令を実行する。この際、ローカルメモリへの書き込みは前述の条件付き命令を用いて各演算器のローカルメモリのPU番地の内容が-1でない演算器のみが実行するように制御している。

【0050】なお、参考までに上記手続きが適用出来る場合を図15、図16に挙げる。図中の13通り全てにおいて上記手順が当てはまることがわかる。

【0051】また、前述の手順S5-13における台形の右斜辺を含むピクセルの面積を計算し、面積の値をローカルレジスタAR0の内容が指し示すローカルメモリのアドレスに書き込む処理は上記の手順(S11)～(S17)と同様である。

【0052】次に、矩形のビットマップ展開の方法について説明する。図17、図18、図19に矩形のビットマップ展開の手順を示す。中間データ(X, Y, N, NBvect, larg1, larg2, larg3, larg4)は既に、各PUのローカルメモリに格納されているものとして説明する。

【0053】(S6-1)変数countを1に初期化する。

(S6-2)矩形の開始座標(X, Y)からピクセル領域のアドレスを計算する。

(S6-3)変数inを1に初期化する。

(S6-4)引き算"count-1"を実行し、ゼロフラグがセットされているPUに限り、変数inにlarg2を代入する(S6-5)。

(S6-5)引き算"NBvect-count"を実行し、ゼロフラグがセットされているPUに限り、変数inにlarg4を代入する(S6-7)。

(S6-8)変数left, rightに各々larg1・in, larg3・inを代入する。

(S6-9)ローカルレジスタAR1にleftを代入する。

(S6-10)ローカルレジスタAR0で指示されるローカルメモリのアドレスにAR1の内容を書き込む。この際、ローカルメモリのアドレスPU番地に-1が書き込まれているPUはローカルメモリへの書込を実行しない。

(S6-11)ローカルレジスタAR0の値をAR0+Nに更新、変数STOPにAR0の値を代入し、ローカルレジスタAR1に変数rightの内容を代入する。

(S6-12)ローカルレジスタAR0で指示されるローカルメモリのアドレスにAR1の内容を書き込む。

(S6-13)ローカルレジスタAR0の値をAR0-Nに更新する。また、ローカルレジスタAR1に変数inの内容を代入する。

(S6-14)ローカルレジスタAR0で指示されるローカルメモリのアドレスにAR1の内容を書き込む。

(S6-15)ローカルレジスタAR0に1を加算し、AR0に代入する。

(S6-16)引き算"STOP-AR0"を実行し、ゼロフラグがセットされてるPUに限り変数PUに-1を代入する(S6-17)。

(S6-18)全てのPUのゼロフラグ、または負フラグがセットされているか否かをテストする。もし、全PUのゼロフラグ、または負フラグがセットされていない場合はS6-14に戻り、セットされている場合は次の処理を実行する。

(S6-19)引き算"NBvect-count"を実行した後にゼロフラグをテストし、ゼロフラグがセットされているPUのみ変数PUに-1を代入する(S6-20)。

(S6-21)全てのPUのゼロフラグ、または負フラグがセットされているか否かをテストする。もし、全PUのゼロフラグ、または負フラグがセットされていない場合は、変数countの値を1増加し(S6-22)、S6-3へ戻り、もし、セットされている場合は次の矩形のビットマップ展開処理を実行する。

【0054】以上説明したように、上記の手順により各PUのローカルメモリ上に矩形の中間データからビットマップが形成される。

【0055】ビットマップ展開処理により得られたビットマップデータは、前述のように各PU間でデータ転送され、ビットマップ比較処理が実行される。

【0056】

【発明の効果】以上のように、この発明の第1の構成によれば、図形要素が座標で表現される画像を、該画像を予め規定した辺長の画素領域に分割し、座標表現された画像データを中間処理して中間データを作成する第1プロセッサ、および複数の演算器と該演算器の演算を制御する制御部とを有するSIMD型プロセッサを含み、該第1プロセッサで作成された中間データを用いて、座標表現された画像データを画素値表現に変換し、第1の画像と第2の画像を比較するサブシステムを備えたので、座標表現の画像データを画素値表現に変換する際においても並列処理ができ、高速に2つの画像を比較検証できる画像処理装置が得られる効果がある。

【0057】また、発明の第2の構成によれば、前記第1プロセッサによる中間処理が、頂点の座標で表現された図形を該図形の境界線を含む画素領域と、その他の画素領域に分類する処理であるので、後続のSIMD型プロセッサでの処理において分岐処理が低減でき、処理効率が向上する。

【0058】また、この発明の第3の構成によれば、前記第1プロセッサによる中間処理が、頂点の座標で表現された台形を底辺と同一方向に走査し、第1の斜辺を含む画素領域と、第2の斜辺を含む画素領域と、その他の画素領域とに分類する処理であるので、台形の図形に対

して第2の構成と同様の効果がある。

【0059】また、この発明の第4の構成によれば、前記第1プロセッサによる中間処理が、頂点の座標で表現された矩形を一方向に走査し、該矩形の境界線を含む画素領域と、その他の画素領域に分類する処理であるので、矩形の図形に対して第2の構成と同様の効果がある。

【0060】また、この発明の第5の構成によれば、前記サブシステムにおいて、SIMD型プロセッサの他に第2プロセッサを備え、該第2プロセッサにより、前記画像中にある規則的に配置する複数の同一図形の繰返し表現を展開処理するようにしたので、第1プロセッサの負荷が軽減されシステムの計算資源がより有効に活用できる。

【0061】また、この発明の第6の構成によれば、前記サブシステムにおいて、SIMD型プロセッサの他に第2プロセッサと該第2プロセッサがアクセスすることの出来るメモリとを備え、該第2プロセッサは、該メモリ上に構成した領域分割テーブルを使用して画像の全部または一部を複数の区画（サブゾーン）に分割し、分割された複数の区画内にある図形要素を各区画毎にSIMD型プロセッサの各演算器に割り当てるようにしたので、図形の座標に従って各図形をまとめる処理が短時間で出来るという効果がある。

【0062】また、この発明の第7の構成によれば、前記SIMD型プロセッサの各演算器に前記分割された区画を割り当てる際に、第1の画像に属する区画とそれに対応する第2の画像に属する区画とを、最もデータ授受の高速な演算器の組に割り当てるようにしたので、各画像に対する画素値の比較が高速に実行出来るという効果がある。

【0063】また、この発明の第8の構成によれば、前記SIMD型プロセッサとして各演算器が環状に接続されたSIMD型プロセッサを用い、第1の画像に属する区画とそれに対応する第2の画像に属する区画を隣接する演算器に割り当てるようにしたので、第7の構成と同様の効果がある。

【0064】また、この発明の第9の構成によれば、前記領域分割テーブルを複数用い、第2プロセッサが一方の領域分割テーブルを使用して区画分割処理を実行中に、並行して他方の領域分割テーブルを用いて前記SIMD型プロセッサに処理させるデータを供給するようにしたので、分割処理と、SIMD型プロセッサで実行する画素値への変換処理とをパイプライン並列処理出来るため、処理効率を向上することが出来る。

【0065】また、この発明の第10の構成によれば、前記SIMD型プロセッサにおいて、各演算器は、演算結果に応じて状態が変化するフラグレジスタを個別に有しており、命令実行の際に、該フラグレジスタに格納されている値が活性状態または非活性状態であることを条

件として、各々の演算器毎に命令を実行するか否かを制御することができ、かつ、全フラグレジスタの値を一度に参照して、全フラグレジスタの値が更新されたか否かを判定することにより全ての演算器が予め定めた条件を満たしたか否かを判定するようにしたので、H/Wコストが軽減できるという効果がある。

【0066】また、この発明の第11の構成によれば、前記SIMD型プロセッサにおいて、各演算器は、少なくとも1個のレジスタと演算結果に応じて状態が変化するフラグレジスタとを個別に有しており、命令実行の際に、予め、画素値表現に必要な所望の座標の値を仮に決定しておき、各該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、該座標の値を更新することにより、該所望の座標を決定し、画素値表現に変換するようにしたので、H/Wコストが軽減できるという効果がある。

【0067】また、この発明の第12の構成によれば、座標で表現される画像データに含まれる台形と予め規定した辺長の画素領域との共有領域の面積を計算することにより画素値表現に変換する際に、該画素領域の境界線と該台形の一方の斜辺との交点の座標値と、該画素領域の頂点の座標値を比較した結果を条件として、予め仮に決定された座標の値を更新するようにしたので、第11の構成と同様の効果がある。

【0068】さらに、この発明の第1の画像処理方法によれば、複数の演算器と該演算器の演算を制御する制御部を有するSIMD型プロセッサにおいて、各演算器は、演算結果に応じて状態が変化するフラグレジスタを個別に有しており、命令実行の際に、該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、各々の演算器毎に命令を実行するか否かを制御することができ、かつ、全フラグレジスタの値を一度に参照して、全フラグレジスタの値が更新されたか否かを判定することにより全ての演算器が予め定めた条件を満たしたか否かを判定するようにしたので、H/Wコストが軽減できるという効果がある。

【0069】さらに、この発明の第2の画像処理方法によれば、複数の演算器と該演算器の演算を制御する制御部を有するSIMD型プロセッサにおいて、各演算器は、少なくとも1個のレジスタと演算結果に応じて状態が変化するフラグレジスタとを個別に有しており、実行命令の際に、予め、変数の値を仮に決定しておき、各該フラグレジスタに格納されている値が活性状態または非活性状態であることを条件として、該変数の値を更新することにより、該変数を決定し、各演算器毎に異なる動作を行わせるようにしたので、H/Wコストが軽減できるという効果がある。

【図面の簡単な説明】

【図1】 この発明の実施の形態1による画像処理装置を示す全体構成図である。

【図2】 この発明の実施の形態1に係わる画像データの一例を示す図である。

【図3】 この発明の実施の形態1に係わる画像データのデータフォーマットを示す図である。

【図4】 この発明の実施の形態1における画像処理システムを示す構成図である。

【図5】 この発明の実施の形態1に係わる台形の中間データを説明する説明図である。

【図6】 この発明の実施の形態1に係わる中間データ生成のための手順を示すフローチャートである。

【図7】 この発明の実施の形態1に係わる矩形の中間データを説明する説明図である。

【図8】 この発明の実施の形態1に係わるサブゾーンについて説明する説明図である。

【図9】 この発明の実施の形態1に係わる第2プロセッサの動作を説明する説明図である。

【図10】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて台形の中間データをビットマップに展開する手順を示すフローチャートである。

【図11】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて台形の中間データをビットマップに展開する手順を示すフローチャートである。

【図12】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて台形の中間データをビットマップに展開する手順を示すフローチャートである。

【図13】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて左斜辺を含むピクセルの値を計算

する手続きを示すフローチャートである。

【図14】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて左斜辺を含むピクセルの値を計算する手続きを説明する説明図である。

【図15】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて左斜辺を含むピクセルの値を計算する手続きを説明する説明図である。

【図16】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて左斜辺を含むピクセルの値を計算する手続きを説明する説明図である。

【図17】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて矩形の中間データをビットマップに展開する手順を示すフローチャートである。

【図18】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて矩形の中間データをビットマップに展開する手順を示すフローチャートである。

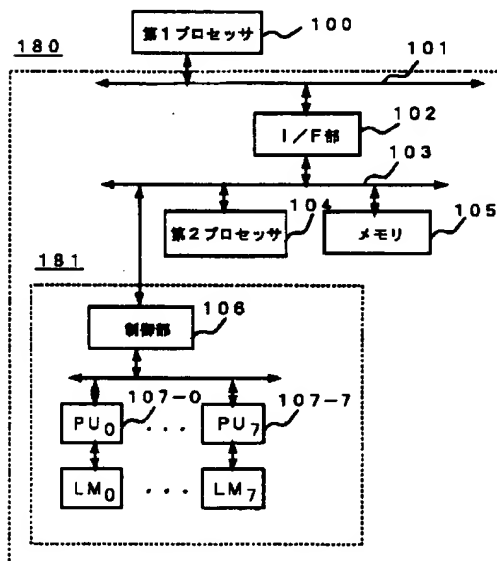
【図19】 この発明の実施の形態1に係わるSIMD型プロセッサにおいて矩形の中間データをビットマップに展開する手順を示すフローチャートである。

【図20】 従来の画像処理装置における画像処理方法を示す説明図である。

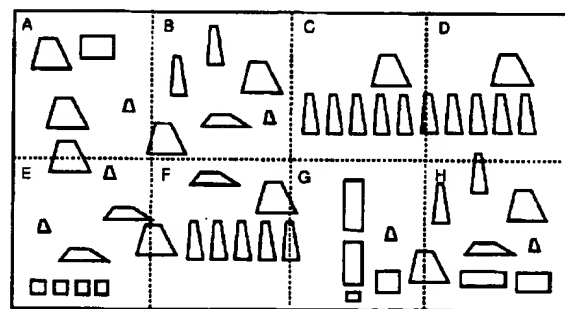
【符号の説明】

100 第1プロセッサ、101 汎用バス、102 I/F部、103 内部バス、104 第2プロセッサ、105 メモリ、106 制御部、107-0~107-7 演算器、180 サブシステム、181 SIMD型プロセッサ。

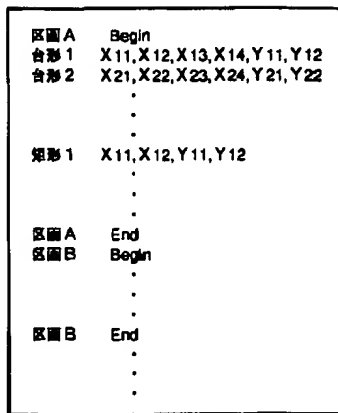
【図1】



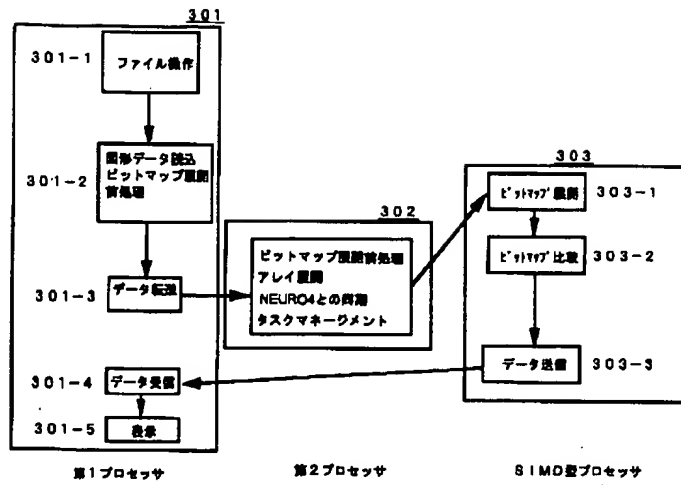
【図2】



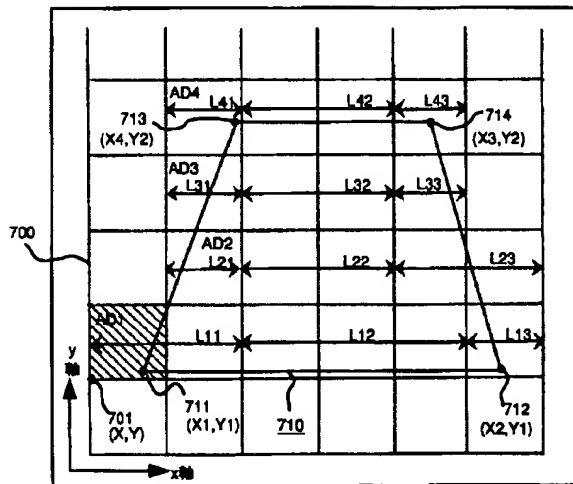
【図3】



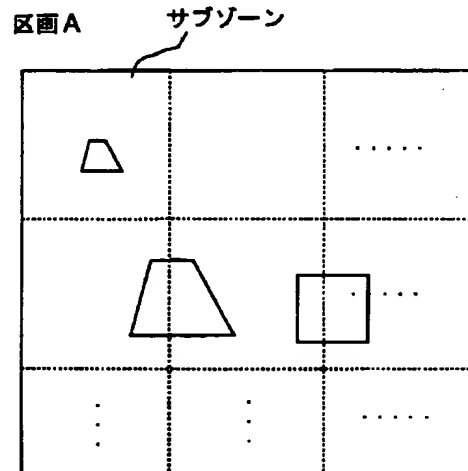
【図4】



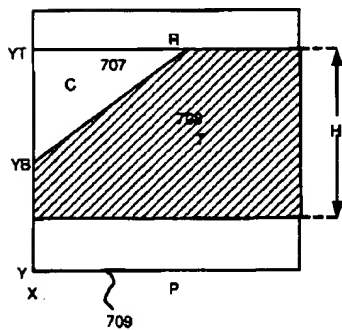
【図5】



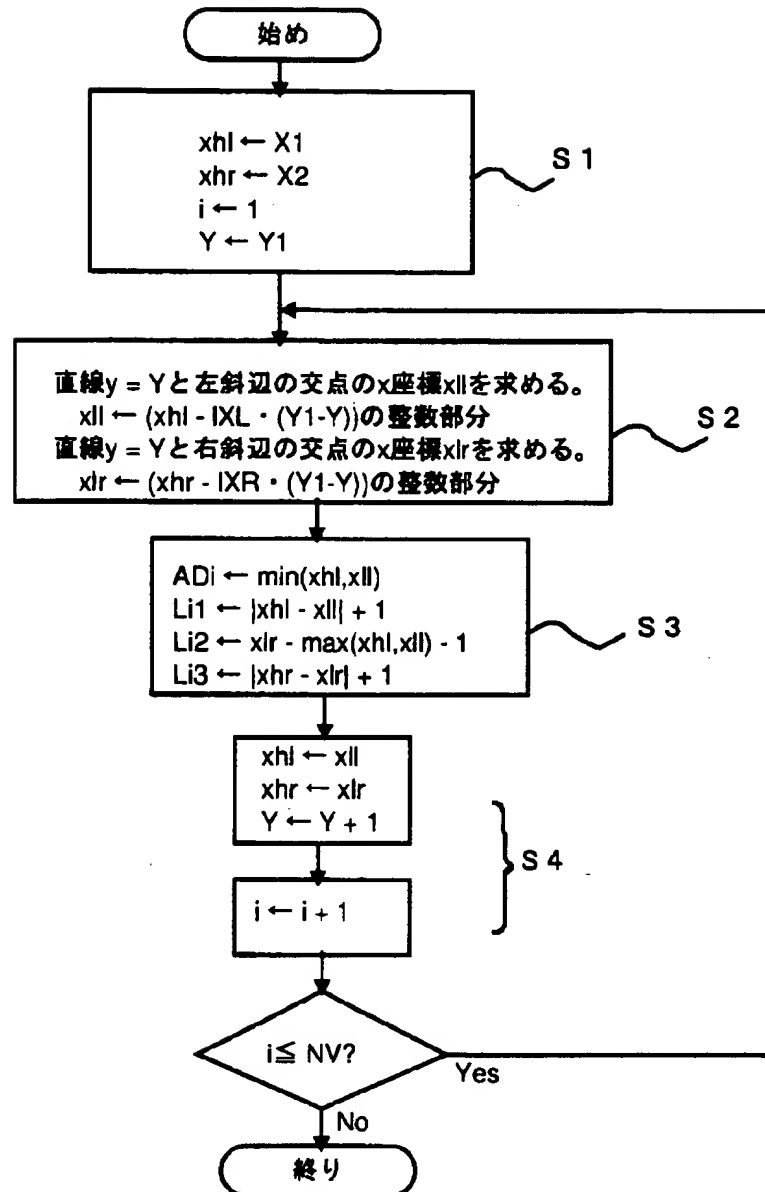
【図8】



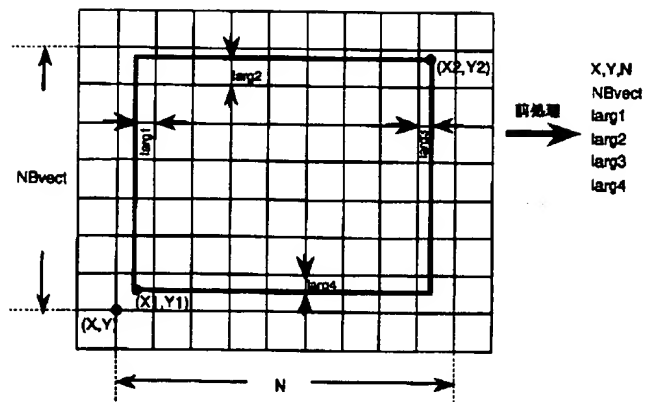
【図14】



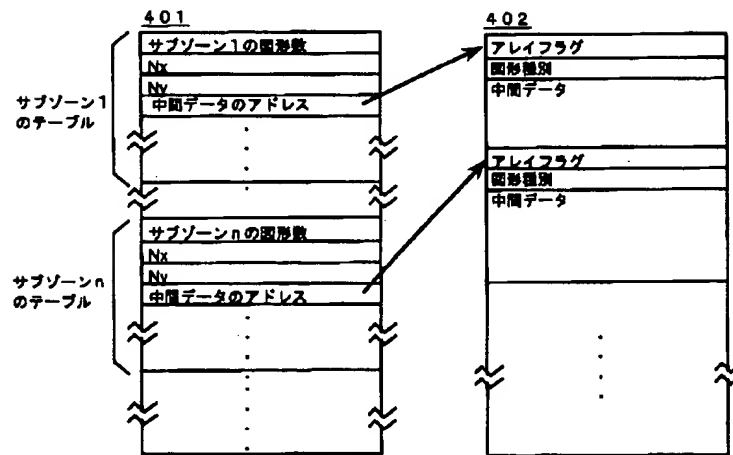
【図6】



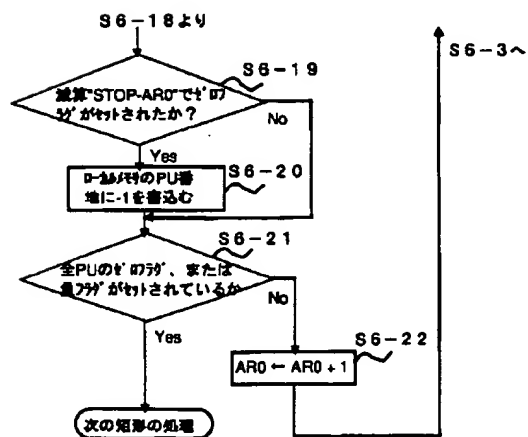
【図7】



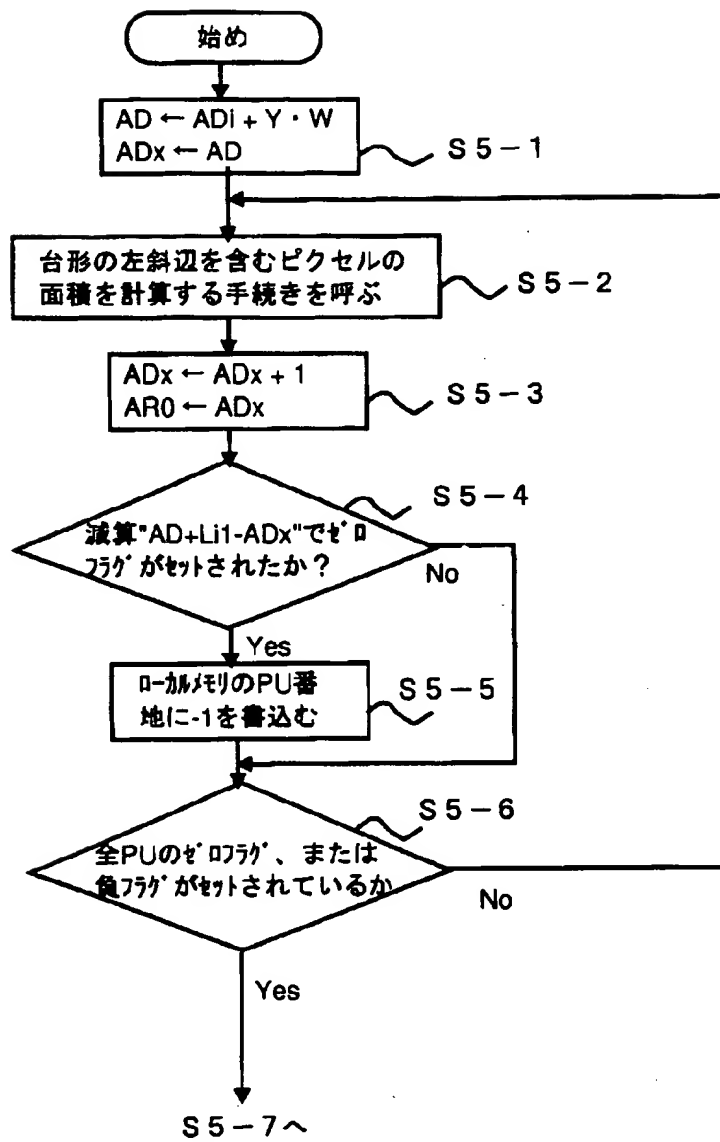
【図9】



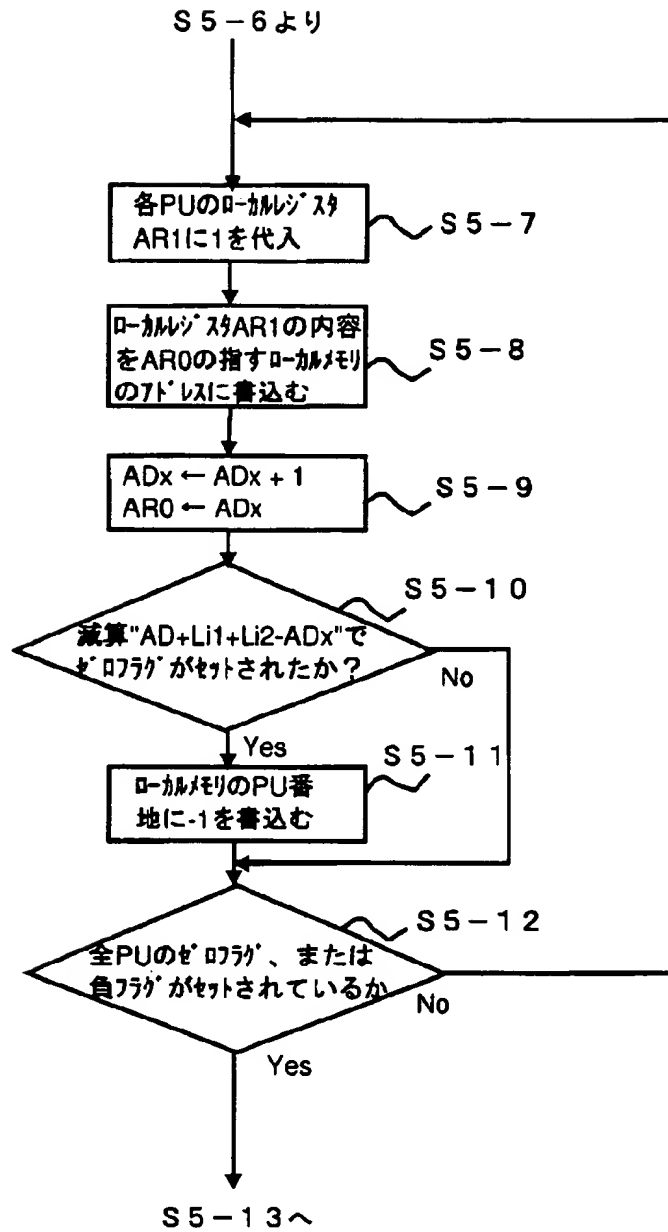
【図19】



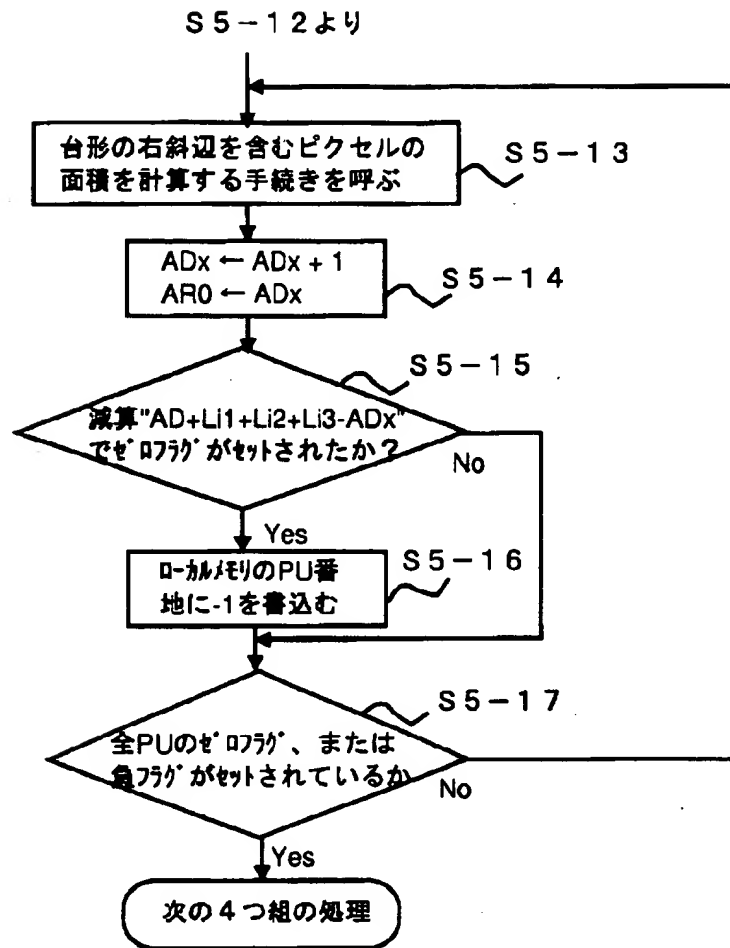
【図10】



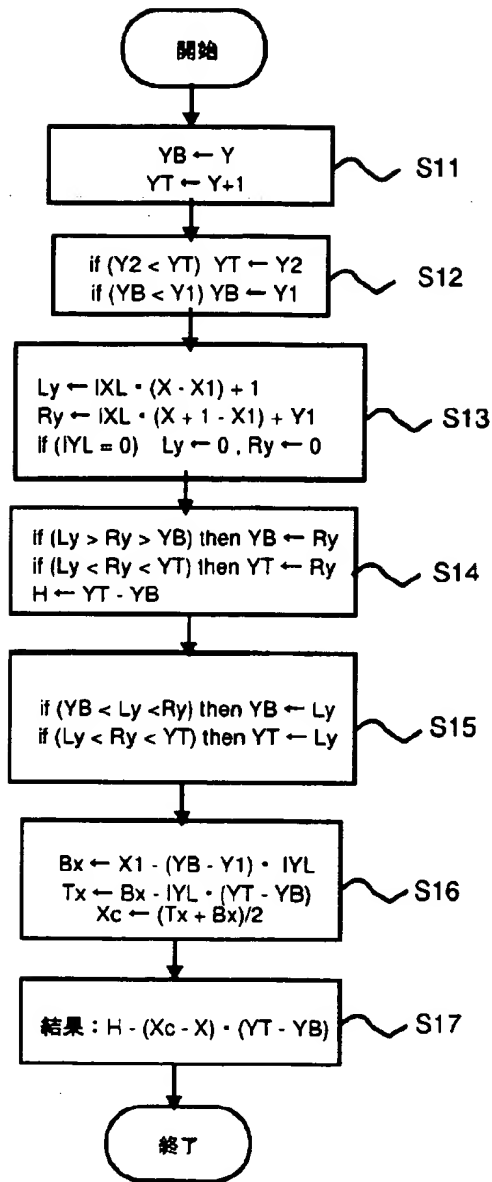
【図11】



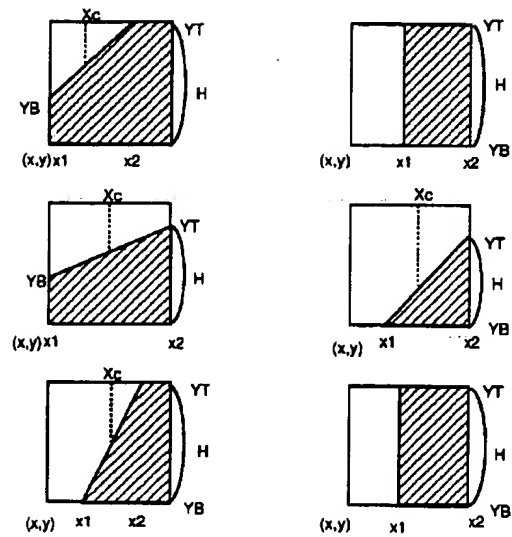
【図12】



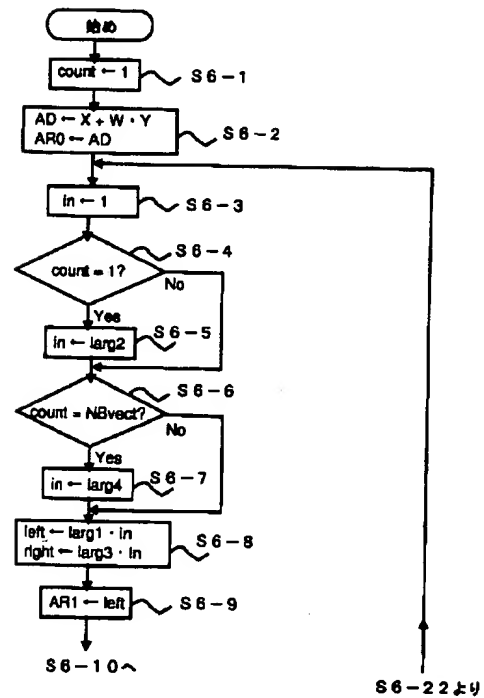
【図13】



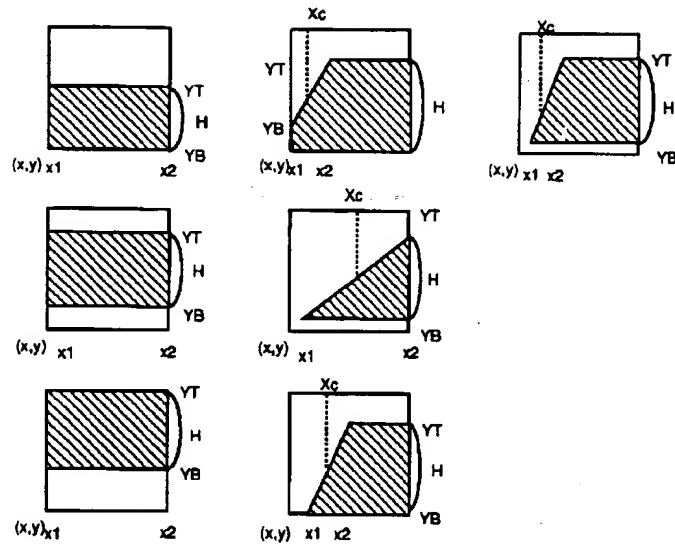
【図16】



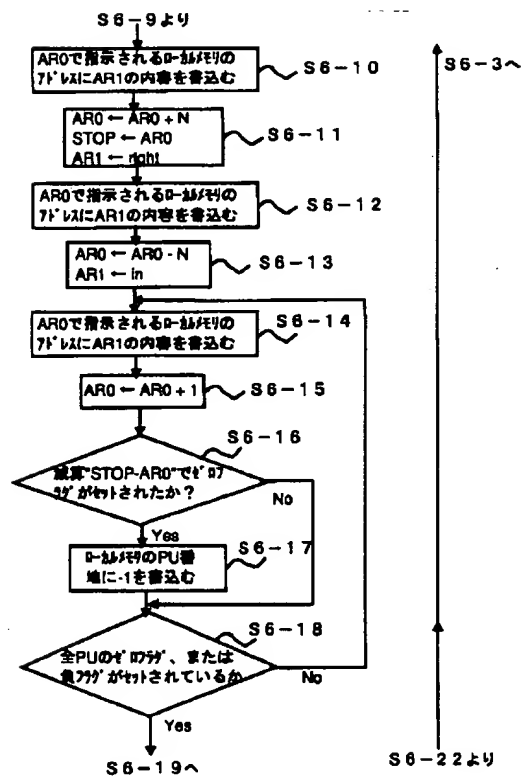
【図17】



【図15】



【図18】



【図20】

